JEE Microservices Configuration

JEE Microservices

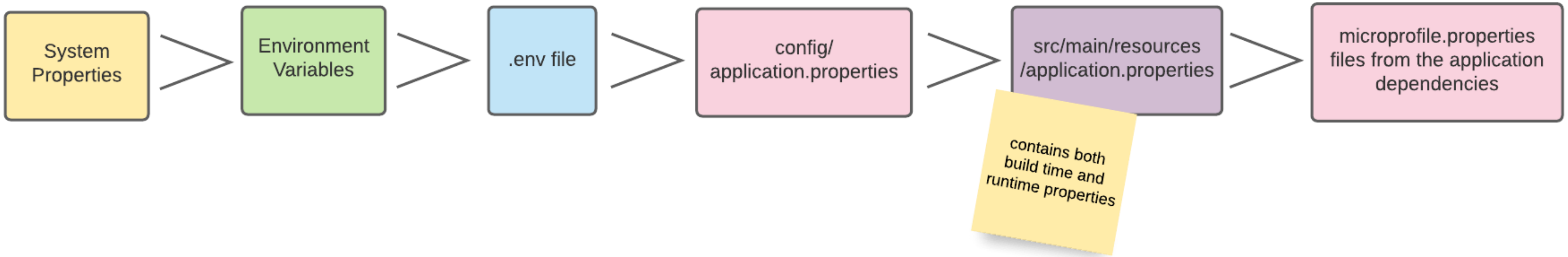
@ CGS IT – 2023

Version 1.0.5

Inhalt

- Quarkus Konfigurations Hierarchie
- Quarkus Config Property Verwendung und Inject
- System Properties
- IntelliJ Run Configuration

Quarkus Config – Quellen und Hierarchie



System Properties > Environment Variables > .env file > config/ application.properties > src/main/resources /application.properties > microprofile.properties files from the application dependencies

contains both build time and runtime properties

Quarkus Config – Quellen und Hierarchie

By default, Quarkus reads configuration properties from multiple sources:

1. (400) System properties
2. (300) Environment variables
3. (295) .env file in the current working directory
4. (260) Quarkus Application configuration file in `$PWD/config/application.properties`
5. (250) Quarkus Application configuration file `application.properties` in classpath
6. (100) MicroProfile Config configuration file `META-INF/microprofile-config.properties` in classpath

https://quarkus.io/guides/config-reference

Use Config Property Example

```
@Path("/helloMicroDemo")
public class DemoMicroResource {
   private static final Logger LOG = Logger.getLogger(DemoMicroResource.class);

   @ConfigProperty(name = "greeting.message", defaultValue = "default-value")
   String message;

   @GET
   @Path("/showMessage")
   @Produces(MediaType.TEXT_PLAIN)
   public String showMessage() {
      LOG.infov(  "INFO  :: showMessage {0}", message);

      try {
         return "Hello: " + message;
      } catch (RuntimeException ex ) {
         LOG.error("fehler beim message lesen ", ex);
      }
      return "";
   }
```

Config Property in Configuration File

```
# Das ist mein erstes Property
greeting.message="greeting configuration  new"


# logging configuration
quarkus.log.console.format=TEE: %d{HH:mm:ss} %-5p [%c{2.}] (%t) %s%e%n
quarkus.log.console.level=INFO
quarkus.log.level=DEBUG
```

Use:

${project rood}/config/application.property
file for external property file

Programatischer Zugriff auf die Configuration

```
String databaseName =
ConfigProvider.getConfig().getValue("database.name",
String.class);

Optional<String> maybeDatabaseName =
ConfigProvider.getConfig().getOptionalValue("database.name",
String.class);
```

Externes File

CGS
ITSolutions

System Properties - Example

System properties can be handed to the application through the -D flag during startup. The following examples assign the value youshallnotpass to the attribute quarkus.datasource.password.

- For Quarkus dev mode:

  ```
  ./mvnw quarkus:dev -Dquarkus.datasource.password=youshallnotpass
  ```

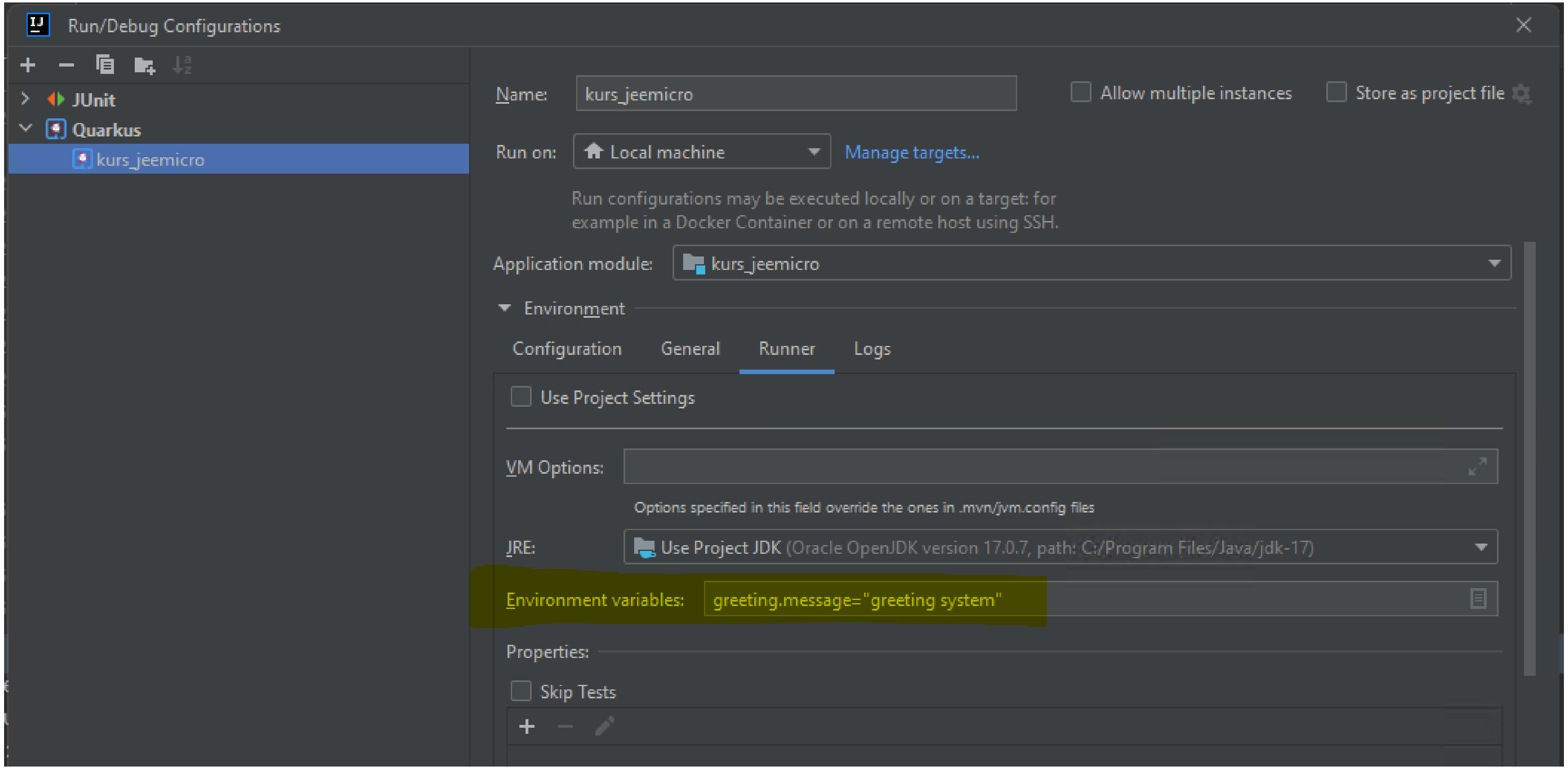
- For a runner jar:

  ```
  java -Dquarkus.datasource.password=youshallnotpass -jar target/quarkus-app/quarkus-run.jar
  ```

- For a native executable:

  ```
  ./target/myapp-runner -Dquarkus.datasource.password=youshallnotpass
  ```

System Variable in IntellIJ



Run/Debug Configurations

- JUnit
- Quarkus
  - kurs_jeemicro

Name: kurs_jeemicro

☐ Allow multiple instances ☐ Store as project file

Run on: Local machine Manage targets...

Run configurations may be executed locally or on a target: for
example in a Docker Container or on a remote host using SSH.

Application module: kurs_jeemicro

Environment

Configuration General Runner Logs

☐ Use Project Settings

VM Options:

Options specified in this field override the ones in .mvn/jvm.config files

JRE: Use Project JDK (Oracle OpenJDK version 17.0.7, path: C:/Program Files/Java/jdk-17)

Environment variables: greeting.message="greeting system"

Properties:

☐ Skip Tests

Danke für Ihre Aufmerksamkeit